Data Engineer Roadmap for 2026

AIRFLOW AWS BEGINNER CAREER DATA ENGINEERING DATA WAREHOUSE GITHUB HADOOP NOSQL PANDAS PYTHON PYTHON SPARK SQL

In the era of data-driven decisions, Data Engineers serve as the backbone, designing the infrastructure that powers modern businesses. They build data pipelines, enable analytics, and turn raw data into valuable insights, making them indispensable yet often overlooked. Whether you're a beginner exploring how to become a Data Engineer or an aspiring Python Data Engineer looking to advance, this article offers a detailed roadmap to navigate this fast-growing field. With rising demand for machine learning and analytics, strong data engineering skills are essential for creating seamless data integration and unlocking the full potential of data. This guide covers key technologies, steps, and best practices to launch a successful career in data engineering.

Learning Objectives

- Understand the role and responsibilities of a data engineer in building robust data infrastructure and enabling data integration across sources.
- Learn the essential programming languages, databases, big data technologies, and cloud platforms required for a career in data engineering.
- Gain hands-on experience with tools and frameworks like Python, SQL, Apache Spark, Hadoop, Apache Kafka, and Apache Airflow through practical projects.
- Develop skills in data warehousing, ETL (Extract, Transform, Load) pipelines, and workflow orchestration for managing complex data engineering tasks.
- Stay updated with the latest trends and emerging technologies in the field, such as real-time data processing, AI/ML integration, and data governance.

Table of contents

- Quarterly Data Engineer Roadmap
- January – Basics of Programming
- February – Fundamentals of Computing
- March – Relational Databases
- April – Cloud Computing Fundamentals with AWS
- May – Data Processing with Apache Spark
- June – Hadoop Distributed Framework
- July – Data Warehousing with Apache Hive
- August – Ingesting Streaming Data with Apache Kafka
- September – Process streaming data with Spark Streaming
- October – Advanced Programming
- November – NoSQL
- December – Workflow Scheduling
- Conclusion

- Frequently Asked Questions

Quarterly Data Engineer Roadmap

Before you jump into the roadmap in detail, it is good to know the quarterly outcome of the roadmap:

- **First Quarter:** Build a strong programming foundation to kickstart your Data Engineering journey.
- **Second Quarter:** Gain hands-on experience in Cloud platforms like Google Cloud Platform (GCP), Microsoft Azure, and tools like Google BigQuery. By the end of this quarter, consider applying for Data Engineering internships.
- **Third Quarter:** Focus on data warehousing and streaming data handling. Master both batch and streaming data pipelines, positioning yourself for entry-level Data Engineering roles.
- **Fourth Quarter:** Dive into Testing, NoSQL Databases, and Workflow Orchestration Tools. By year-end, you'll have a comprehensive grasp of essential Data Engineering tools and cloud services like GCP, ready to ace job interviews.

January – Basics of Programming

The first thing you need to master to become a Data Engineer in 2025 is a programming language. This will kickstart your journey in this field and allow you to think in a structured manner. And there is no better programming language to start your programming journey with than Python.

Python is one of the most suitable programming languages for Data Engineering. It is easy to use, has a dearth of supporting libraries, has a vast community of users, and has been thoroughly incorporated into every aspect and tool of Data Engineering.

- Understanding Operators, Variables, and Data Types in Python
- Conditional Statements & Looping Constructs
- Data Structures in Python (Lists, Dictionaries, Tuples, Sets & String methods)
- Writing custom Functions (incl. lambda, map & filter functions)
- Understanding Standard Libraries in Python
- Using basic Regular Expressions for data cleaning and extraction tasks

Besides this, you should specifically focus on the **Pandas library in Python**. This library is widely used for reading, manipulating, and so on. Here you can focus on the following:

- Basics of data manipulation with Pandas library
- Reading and writing files with Pandas
- Manipulate columns in Pandas – Rename columns, sort data in Pandas dataframe, binning data using Pandas, etc.
- How to deal with missing values using Pandas
- Apply function in Pandas
- **Pivot table**
- **Group by**

February – Fundamentals of Computing

Once you become comfortable with the Python programming language, it is essential to focus on some computing fundamentals to become a Data Engineer in 2025. This is extremely helpful because, most of the time, the data sources you are working with will require you to grasp these computing fundamentals well.

It would help if you focused on **shell scripting in Linux** as you will be working with the Linux environment in a shell format. You will extensively be using shell scripting for cron jobs, setting up environments, or even working in the distributed environment, which is used widely by Data Engineers. Besides this, you will need to work with APIs. This is important because projects involve multiple APIs that send and receive data. So learn the basics of APIs like GET, PUT, and POST. The requests library is widely used for this purpose.

Web Scraping is also essential for a Data Engineer's day-to-day tasks. This is because, a lot of the time, we need to extract data from websites that might not have a straightforward or helpful API. For this, you can focus on working with BeautifulSoup and Selenium libraries. Finally, master Git and GitHub as they are handy for version control. Different members will be working on a single project when you are working in a team. Without a version control tool, this collaboration would be impossible.

March – Relational Databases

No Data Engineering project is complete without a storage component. A relational Database is one of the core storage components used widely in Data Engineering projects. One needs a good understanding of relational databases like MySQL to work with the enormous amounts of data generated in this field. Relational databases are widely used to store data in any given field. This is because of their ACID properties, which allow them to handle transactional data easily.

Having a solid grasp of MySQL, including skills like writing complex SQL queries, understanding database normalization, indexing, and performance optimization, is crucial for a Data Engineer. The ability to design and manage efficient MySQL databases is a fundamental requirement in many data engineering roles and projects.

To work with relational databases, you must master the Structure Query Language (SQL). You can focus on the following while learning SQL:

- Basic Querying in SQL
- **Keys in SQL**
- **Joins in SQL**
- Practice Subqueries in SQL
- Constraints in SQL
- **Window Functions**
- Normalisation

Project

At the end of the first quarter, you will understand programming, SQL, web scraping, and APIs well. This should give you enough leeway to work on a small sample project.

In this project, you can focus on bringing in data from any open API or scraping the data from a website. Transform that particular data using Pandas in Python. And finally, store it in a relational database. A similar project can be found here, where we analyze streaming tweets using Python and PostgreSQL.

April – Cloud Computing Fundamentals with AWS

We will start this quarter with a focus on cloud computing because, given the size of Big Data, Data Engineers find it extremely useful to work on the cloud. This allows them to work with the data without resource limitations. Also, given the proliferation of cloud computing technologies, it has become increasingly easier to manage complex work processes entirely on the cloud.

- Learn the basics of AWS
- Learn about IAM users and IAM Roles
- Learn to launch and operate an EC2 on AWS
- Get comfortable with Lambda Functions on AWS
- AWS S3 is the primary storage component
- API gateway
- Practice networking with AWS VPC
- Practice databases with AWS RDS and Aurora

Also Read: **What is AWS | Amazon Web Services for Data Science**

Project

You can take your project from the previous quarter to the cloud by following the below steps:

- Use API Gateway to ingest the data from Twitter API
- Process this data with AWS Lambda
- Store the processed data in AWS Aurora for further analysis

May – Data Processing with Apache Spark

Next, you need to learn how to process Big Data and handle large datasets effectively, which is a crucial skill for data analysis and data analytics roles. Big Data broadly has two aspects, batch data, and streaming data. This month, you should focus on learning the tools to handle batch data. Batch data is accumulated over time, say a day, a month, or a year. Since the data is extensive in such cases, we need specialized tools. One such popular tool is Apache Spark.

You can focus on the following while learning **Apache Spark**:

- Spark architecture
- RDDs in Spark
- Working with Spark Dataframes
- Understand Spark Execution
- Broadcast and Accumulators
- Spark SQL

While you are at it, also learn about the ETL (Extract, Transform, Load) pipeline concept. ETL is nothing but Extracting the data from a source, Transforming the incoming data into the required format, and finally

Loading it to a specified location. Apache Spark is widely used for this purpose, and Data Engineers use ETL pipelines in every project involving large datasets!

You can work with the **Databricks community edition** to practice Apache Spark.

June – Hadoop Distributed Framework

No Data Engineering project is complete without utilizing the capabilities of a distributed framework. Distributed frameworks allow Data Engineers to distribute the workload onto multiple small-scale machines instead of relying on a single massive system. This provides higher scalability and better fault tolerance.

- Get an overview of the Hadoop Ecosystem
- Understand MapReduce architecture
- Understand the workings of YARN
- Work with Hadoop on the cloud with **AWS EMR**

Project – By the end of this quarter, you will have a good understanding of handling batch data in a distributed environment, and you will also have the basics of cloud computing in place. Also, you can start your data engineering journey with these tools by applying to Data Engineering internships.

To showcase your skills, you can build a project on the cloud. You can take up any datasets from the **DataHack platform** and practice working with the Spark framework.

July – Data Warehousing with Apache Hive

Getting data into databases is only one-half of the work. The real challenge is aggregating and storing the data in a central repository. This will act as a single version of the truth that anyone from an organization can query and get a common and consistent result. You will first need to understand the differences between Database, Data Warehouse, and Data lake since you will often come across these terms. Not only this but also try to understand the difference between OLTP vs OLAP.

Next, you should focus on the modeling aspect of data warehouses. To learn about the Star and Snowflake schema often used in designing data warehouses. Finally, you can start to learn the various data warehouse tools. One of the most popular is Apache Hive which is built on top of Apache Hadoop and is used widely in the industry. While learning Hive, you can focus on the following topics:

- Hive Query Language
- Managed vs External tables
- Partitioning and Bucketing
- Types of File formats
- SerDes in Hive

Once you have mastered the basics of Apache Hive, you can practice working with it on the cloud using the AWS EMR service.

August – Ingesting Streaming Data with Apache Kafka

Having worked extensively with batch data, it is time to move on to streaming data.

Streaming data is the data that is generated in real time. Think of tweets being generated, clicks being recorded on a website, or transactions occurring on an e-commerce website. Such data sometimes needs to be handled in real time. For example, we may need to determine in real-time whether a tweet is toxic so we can easily remove it from the platform. Or determine whether a transaction is fraudulent or not to prevent it from causing extensive damage.

The problem with working with such data is that we must ingest it in real time and process it at the same rate. This will ensure that there is no data loss in the interim. To ensure that data is being ingested reliably while it is being generated, we need to use Apache Kafka.

- Learn Kafka architecture
- Learn about Producers and Consumers
- Create topics in Kafka

Besides this, you can also learn about **AWS Kinesis** which is also used to handle streaming services on the cloud.

September – Process streaming data with Spark Streaming

Once you have learned to ingest streaming data with Kafka, it is time to learn how to process that data in real-time. Although you can do that with Kafka, it is not as flexible for ETL purposes as Spark Streaming. Spark Streaming is part of the core Spark API. You can focus on the following while learning Spark Streaming:

- DStreams
- Stateless vs. Stateful transformations
- Checkpointing
- Structured Streaming

Project

At the end of this quarter, you will be able to handle batch and streaming data. You will also have a good knowledge of data warehousing. So, you will know most of the tools and technologies a data engineer needs to master.

To showcase your skills, you can build a small project. Here you can leverage the benefits of streaming services in your Twitter sentiment analysis project from before. Let's say you want to analyze all the tweets related to Data Engineering in real time and store them in a database.

- You ingest tweets from API into AWS Kinesis. This will ensure that you are not losing out on any incoming tweets.
- These tweets can be processed in small batches on AWS EMR with Spark Streaming API.
- The processed tweets can be stored in Hive tables.
- These processed tweets can be aggregated daily using Spark. For example, you can find the total number of tweets, common hashtags used, any upcoming event highlighted in the tweets, and so on.

October – Advanced Programming

```c
/*
 * This line basically imports the "stdio" header file, part of
 * the standard library. It provides input and output functionality
 * to the program.
 */
#include <stdio.h>

/*
 * Function (method) declaration. This outputs "Hello, world\n" to
 * standard output when invoked.
 */
void sayHello(void) {
    // printf() in C outputs the specified text (with optional
    // formatting options) when invoked.
    printf("Hello, world!\n");
}

/*
 * This is a "main function". The compiled program will run the code
 * defined here.
 */
int main(void)
{
    // Invoke the sayHello() function.
    sayHello();
    return 0;
}
```

Source: Wikipedia

The final quarter will focus on mastering the skills required to complete your data engineering profile. We will start by bringing our attention back to programming. We will focus on some advanced programming skills, which will be vital to your growth as a Data Engineer in 2025. These are useful for working on a larger project in the industry as we need to incorporate the best programming practices. For this, you can work on the following:

- **OOPs concepts** – Classes, Objects, Inheritance
- Understand Recursion functions
- Testing – Unit and Integration testing

November – NoSQL

Having worked with relational databases, you must have noticed some glaring drawbacks like the data always needs to be structured, the querying is not that fast when working with relatively large data, and even it has scalability issues. To overcome such drawbacks, the industry came up with NoSQL databases. These databases can deal with structured and unstructured data, are quick for data insertions, and even the querying is much faster. They are increasingly being used in the industry to capture user data.

So start by understanding the difference between SQL and NoSQL databases. Once you have done that, focus on the different types of NoSQL databases.

You can focus on learning one particular NoSQL database. I would suggest going for MongoDB as it is popularly used in the industry and is also very easy to learn for someone who has already mastered SQL. To learn MongoDB, you can focus on the following :

- **CAP theorem**
- Documents and Collections

- CRUD operations
- Working with different types of operators – Array, Logical, Comparison, etc.
- Aggregation Pipeline
- Sharding and Replication in MongoDB

Project

As an excellent hands-on practice, I would encourage you to set up a MongoDB cluster on AWS. You will need to host MongoDB servers on different EC2 instances. You can treat one of these as the Primary, and the rest can be treated as Secondary nodes. Further, you can employ Sharding and Replication concepts to solidify your understanding. And you can use any open-source API to treat it as a source of incoming data.

This is no doubt going out of the way and exploring NoSQL in-depth, but it will give you a good understanding of how the servers are set up and interact with each other in the real world and will be a bonus towards your journey of becoming a Data Engineer in 2025.

December – Workflow Scheduling

The ETL pipelines you build to get the data into databases and data warehouses must be managed separately. This is because any Data Engineering project involves building complex ETL pipelines, which must be scheduled at different time intervals and work with varying data types. We need a powerful workflow scheduling tool to manage such pipelines successfully and gracefully handle the errors. One of the most popular ones is **Apache Airflow**.

You can learn the following concepts in Apache Airflow:

- DAGs
- Task dependencies
- Operators
- Scheduling
- Branching

Project

By now, you will have a thorough understanding of the essential tools in data engineering. These will prepare you for your dream job interview and to become a Data Engineer in 2025.

To showcase your skills, you can build a capstone project. Here, you can take up batch and streaming data to showcase your holistic knowledge of the field. You can manage the ETL pipelines using Apache Airflow. And finally, think of taking the project onto the cloud so that there is no resource crunch at any point in time. This was the final leap of faith in becoming a Data Engineer in 2025.

Conclusion

These tools will get your journey started in the Data Engineering field. Armed with these tools and sample projects, you can quickly nail any internship and job interview and become a Data Engineer in 2025. But

these Data Engineering tools are constantly evolving, so it's crucial to stay updated with the most recent technologies and trends related to data storage, data processing, and data quality assurance.

Platforms like LinkedIn can be invaluable for networking with data professionals and learning about new tools and best practices. Additionally, pursuing relevant certifications from cloud providers or open-source foundations can further validate your skills and make you a more competitive candidate for Data Engineering roles in 2024. Happy learning and best of luck on your data engineering journey!

Key Takeaways

- Data engineers play a crucial role in constructing the foundations for data-driven businesses by shaping data pipelines and enabling analytics.
- The roadmap covers a comprehensive set of skills, including programming, cloud computing, big data processing, data warehousing, and workflow management.
- Mastering tools like Apache Spark, Hadoop, Kafka, Hive, and Airflow is essential for handling batch and streaming data, data warehousing, and orchestrating complex ETL pipelines.
- Hands-on projects and internships are vital for gaining practical experience and showcasing skills to potential employers.
- Continuous learning and staying updated with evolving technologies, such as AI/ML integration and real-time data processing, are necessary to excel as a data engineer.

Also Read: [Learning Path to Become a Data Analyst in 2025]

Frequently Asked Questions

Q1. What is the roadmap for a data engineer?

A. The roadmap for a data engineer involves mastering skills like SQL, ETL, data modeling, progressing to cloud platforms, big data tech, and programming. Gain experience with real projects and stay updated with evolving tech.

Q2. How do I become a data engineer?

A. To become a data engineer, start with programming (Python/Java) and SQL. Learn ETL techniques, databases, big data tools, and cloud platforms. Gain hands-on experience through internships or personal projects, and keep learning and networking.

Q3. What's next for data engineers?

A. The future of data engineering includes AI and ML integration, real-time data processing, security, and governance. Data engineers should adapt to these trends, stay updated, and expand skills in these areas.

Q4. What should I do to become a data engineer?

A. Steps to become a data engineer: Learn SQL, programming, ETL, data warehousing, big data tech, and cloud platforms. Apply knowledge through projects, seek internships, and continuously learn about emerging technologies in the field.

Article Url - https://www.analyticsvidhya.com/blog/2023/01/step-by-step-roadmap-to-become-a-data-engineer/

Aniruddha Bhandari

I am on a journey to becoming a data scientist. I love to unravel trends in data, visualize it and predict the future with ML algorithms! But the most satisfying part of this journey is sharing my learnings, from the challenges that I face, with the community to make the world a better place!